《KERMIT: BERT 的逻辑化》

2.0 修正版

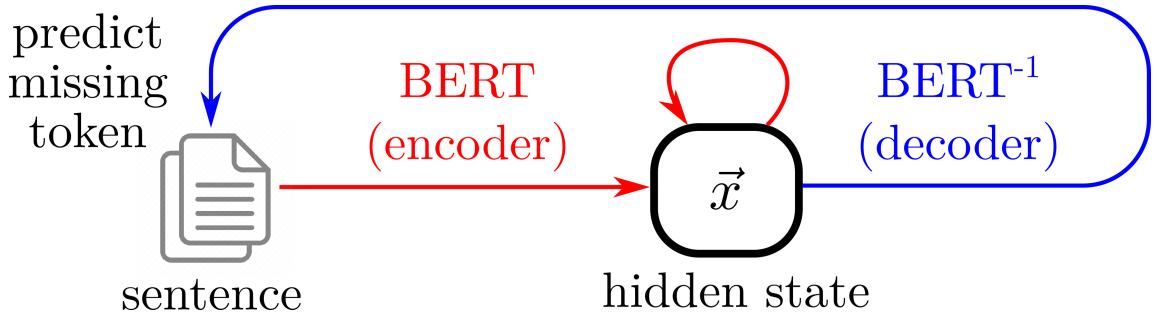
YKY

August 16, 2020

Table of contents

BERT 的革命性意义：「闭环路训练」

- BERT 利用平常的文本 induce 出知识，而这 representation 具有 通用性 (universality)：



(1)

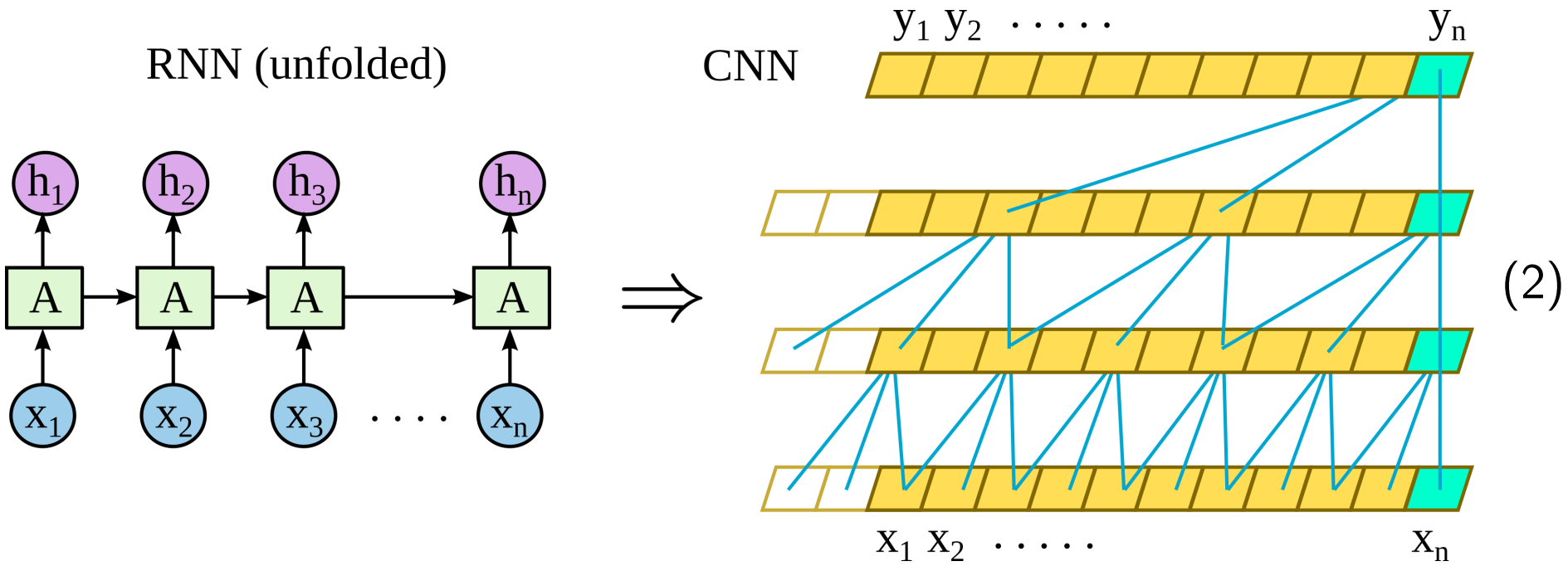
换句话说：隐状态的 representation 压缩了句子的意思，而它可以应用在别的场景下

- This implies that human-level AI can be *induced* from existing corpora, 而不需重复 像人类婴儿成长的学习阶段
- 这种训练方法是较早的另一篇论文提出，它并不属於 BERT 的内部结构

BERT 的内部结构

其实，BERT 也是混合了很多技巧 发展而成的：

- BERT 基本上是一个 seq-to-seq 的运算过程
- Seq-to-seq 问题最初是用 RNN 解决的
- 但 RNN 速度较慢，有人提出用 CNN 取代：



- CNN 加上 attention mechanism 变成 Transformer
- 我的想法是重复这个思路，但引入 逻辑的对称性

Symmetry in logic

- 词语 组成 句子，类比於 逻辑中，概念 组成 逻辑命题
- 抽象地说，逻辑语言 可以看成是一种有 2 个运算的 代数结构：加法（$\wedge$，合并命题，可交换）和 乘法（$\cdot$，用作概念合成，不可交换）
- 例如：

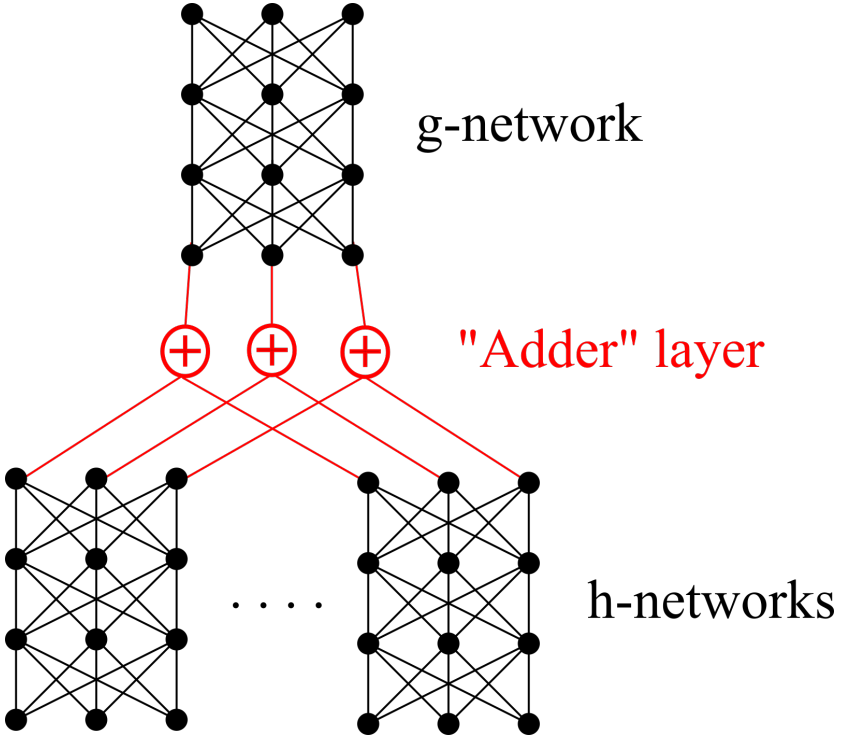
$$\begin{aligned} A \wedge B \quad &\equiv \quad B \wedge A \\ \text{下雨} \wedge \text{失恋} \quad &\equiv \quad \text{失恋} \wedge \text{下雨} \end{aligned} \tag{3}$$

- Word2Vec 也是革命性的；由 Word2Vec 演变成 Sentence2Vec 则比较容易，基本上只是 向量的 合并 (concatenation); Sentence 对应於 逻辑命题
- 但 命题的 集合 需要用 symmetric NN 处理，因为 集合的元素 是顺序无关的

Symmetric neural network

- Symmetric NN 问题 已经由 两篇论文解决了:
  [PointNet 2017] [DeepSets 2017]
- Any symmetric function can be represented by the following form (a special case of the Kolmogorov-Arnold representation of functions):

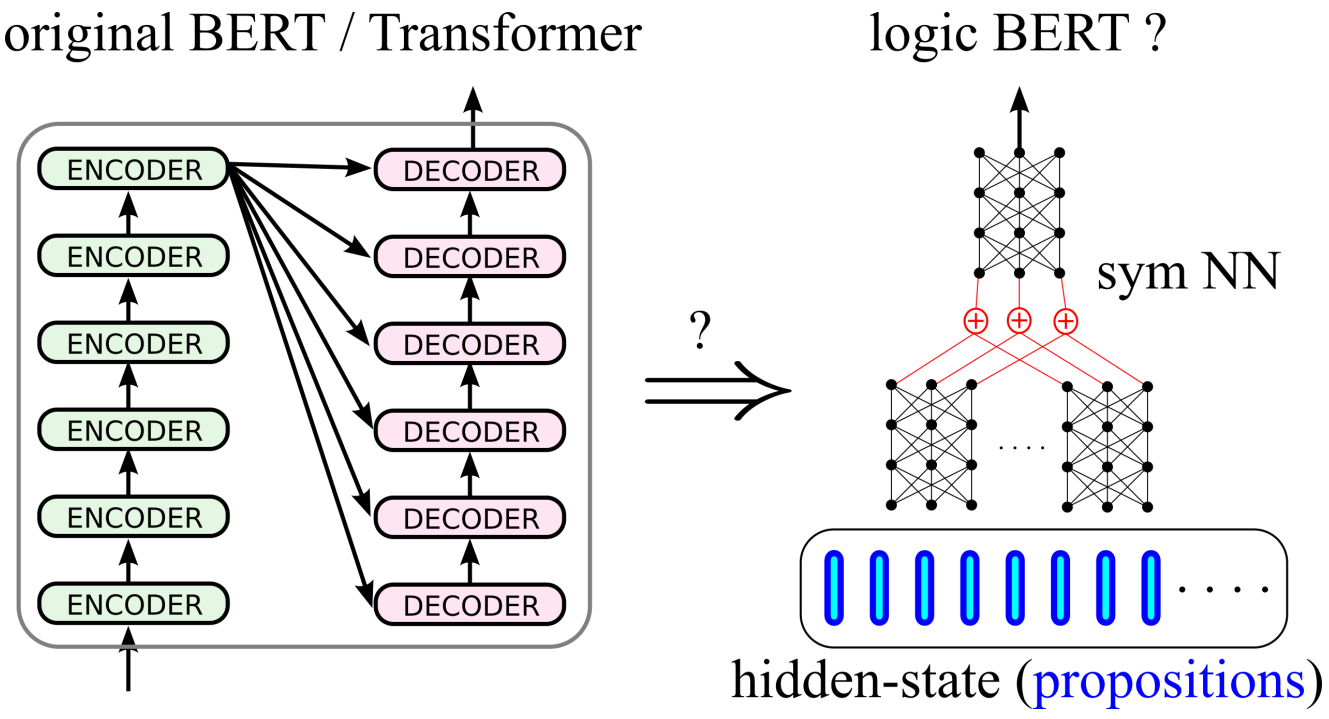
$$f(x, y, ...) = g(h(x) + h(y) + ...) \tag{4}$$



(5)

BERT 的逻辑化

- 可以强逼 BERT 的 隐状态 变成 “set of propositions” 的形式，方法是将对称性 施加在 Encoder 上：



(6)

- 下面会看到，其实这并无需要，因为 BERT 的「注意力机制」已经是对称的，它可以做 逻辑推导

逻辑 与 AI 之间的联系

- 既然 AI 基於 逻辑，则 AI 与逻辑之间 必然存在 精确 (precise) 的联系
- BERT 似乎是在执行 句子之间的变换，而这些句子是 word embedding 的 concatenation，例如：

$$\text{苏格拉底} \cdot \text{是} \cdot \text{人} \xrightarrow{BERT} \text{苏格拉底} \cdot \text{会} \cdot \text{死} \tag{7}$$

这个做法看似很「粗暴」，其实它和 逻辑式子 的作用一样：

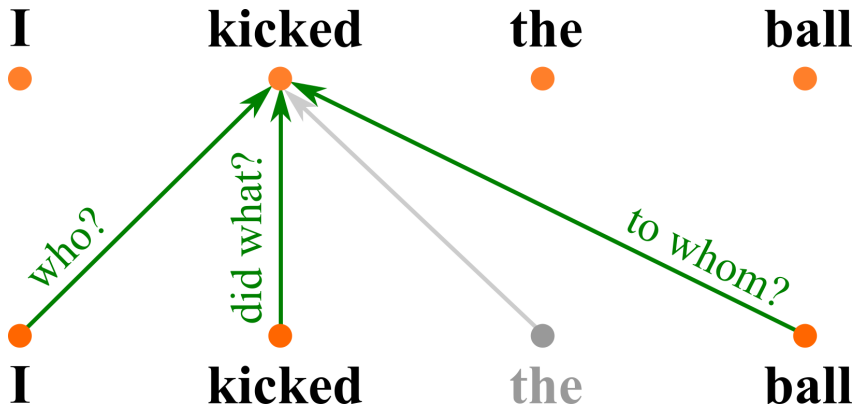
$$\forall x.\ \mathsf{Human}(x) \to \mathsf{Mortal}(x) \tag{8}$$

而这式子，根据 Curry-Howard 对应，就是 (7) 的函数映射!

- 我会在另一辑 slides 里 简介一下这些理论；可以说，逻辑的 几何结构 是「永恒」的，它可以指示 AI 的 长远发展

Attention 是什么?

- 注意力 最初起源於 Seq2seq，后来 BERT 引入 self-attention
- Attention 的本质就是 加权，权值 可以反映 模型 关注的点
- For each input, attention weighs the relevance of every other input and draws information from them accordingly to produce the output
- 在 BERT 里，attention 是一种 words 之间的关系:

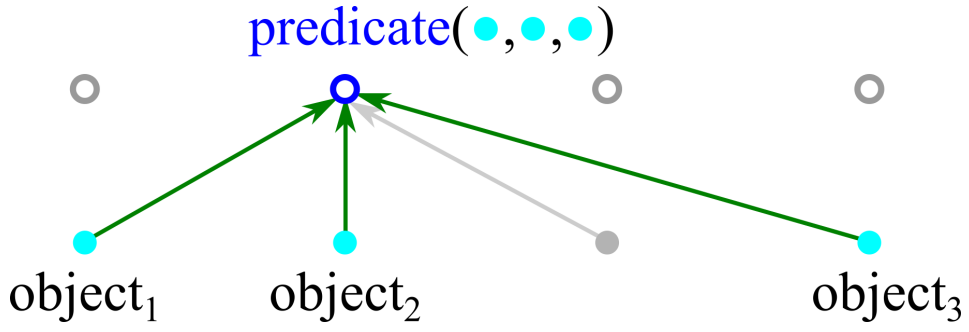


(9)

- 但，从逻辑的角度看，word $\neq$ 命题
- 在逻辑学上，必需分清 命题内部 与 命题之间 这两个层次，非常关键!

谓词 (predicates) vs 命题 (propositions)

- “Predicate” 来自拉丁文「断言」的意思
- 逻辑里，predicate 代表一个 没有主体 / 客体 的断言，换句话说，是一个有「洞」的命题
- 命题 = 谓词 (predicate) + 主体 / 客体（统称 objects）
- 例如：Human(John), Loves(John, Mary)
- 从逻辑的角度看，attention 的输出可以看成是 predicate 和 objects 的结合：



(10)

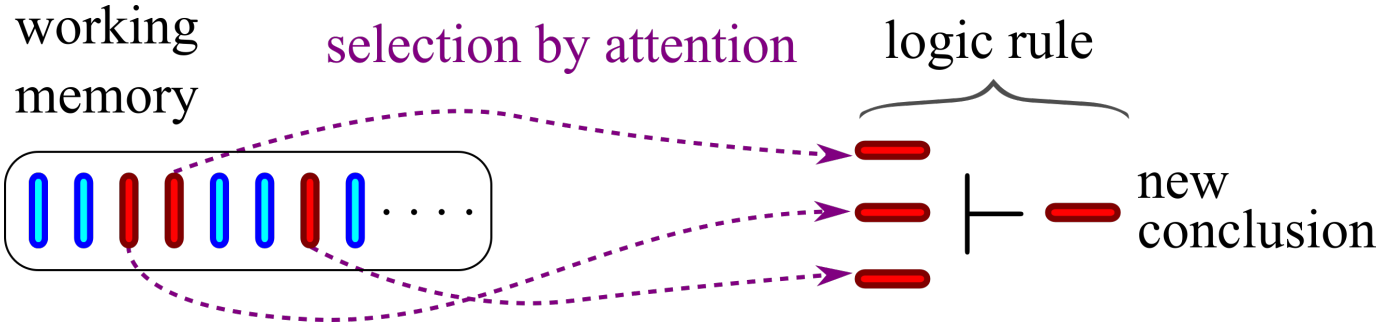
- 形象地说：

$$\text{predicate} + \text{objects} = \text{proposition} \tag{11}$$

$$\circ + \bullet\bullet\bullet \ldots = \text{—}$$

“Attention is all you need” ?

- 类似地，高层的 attention 可以处理 命题之间 的关系
- 我们希望 attention 做到的是 选择有关联的命题，去做逻辑推导：



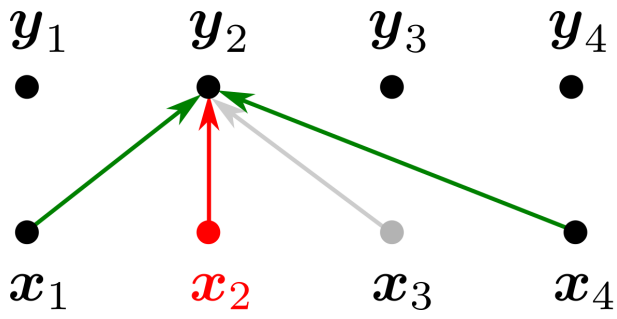
(12)

- 但从 N 个命题中选择 K 个，可以有 $\binom{N}{K}$ 个子集，是 exponential 的
- BERT 的做法是：每次只输出 N 个命题，而前提的 “support” 也是上一层的 全部 N 个命题，每个前提的「影响力」由 matrix 权重决定
- 根据 Curry-Howard isomorphism，BERT 的映射 其实对应於某种 另类的逻辑（BERT 的设计者可能没有意识到这点），这种逻辑的好处是运行非常快
- BERT 的逻辑 看似有局限，但这种表面的局限未必阻止它是 universal 的逻辑
- 关键是在 速度 与 逻辑的 expressive power 之间 找到平衡

- 最简单的 attention 公式是（其中 Q, K, V = query, key, value 矩阵）：

$$\boldsymbol{y}_j = \sum_i \langle Q\boldsymbol{x}_j, K\boldsymbol{x}_i \rangle V\boldsymbol{x}_i \tag{13}$$

（红色 代表注意力的 focus）这对应於一个逻辑式子：



$$\equiv \quad \boldsymbol{x}_1 \wedge \boldsymbol{x}_2 \wedge \boldsymbol{x}_3 \wedge \boldsymbol{x}_4 \vdash \boldsymbol{y}_2 \tag{14}$$

亦即是说：$\boldsymbol{y}_j$ 是由 $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$ 得出的逻辑结论 with focus on $\boldsymbol{x}_j$

- 所谓 “focus” 并不是 逻辑概念，它只是 BERT 加速的 heuristic
- 容易看到，attention 对於 $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$ 是 交换不变的 (equivariant)，这表示每层 attention 的输出是一些 逻辑命题，和我的理论相符
- Multi-head attention 亦有一个很好的逻辑解释：即使 focus $= \boldsymbol{x}_j$，亦有其他不同的前提，可以导致不同的结论，例如：

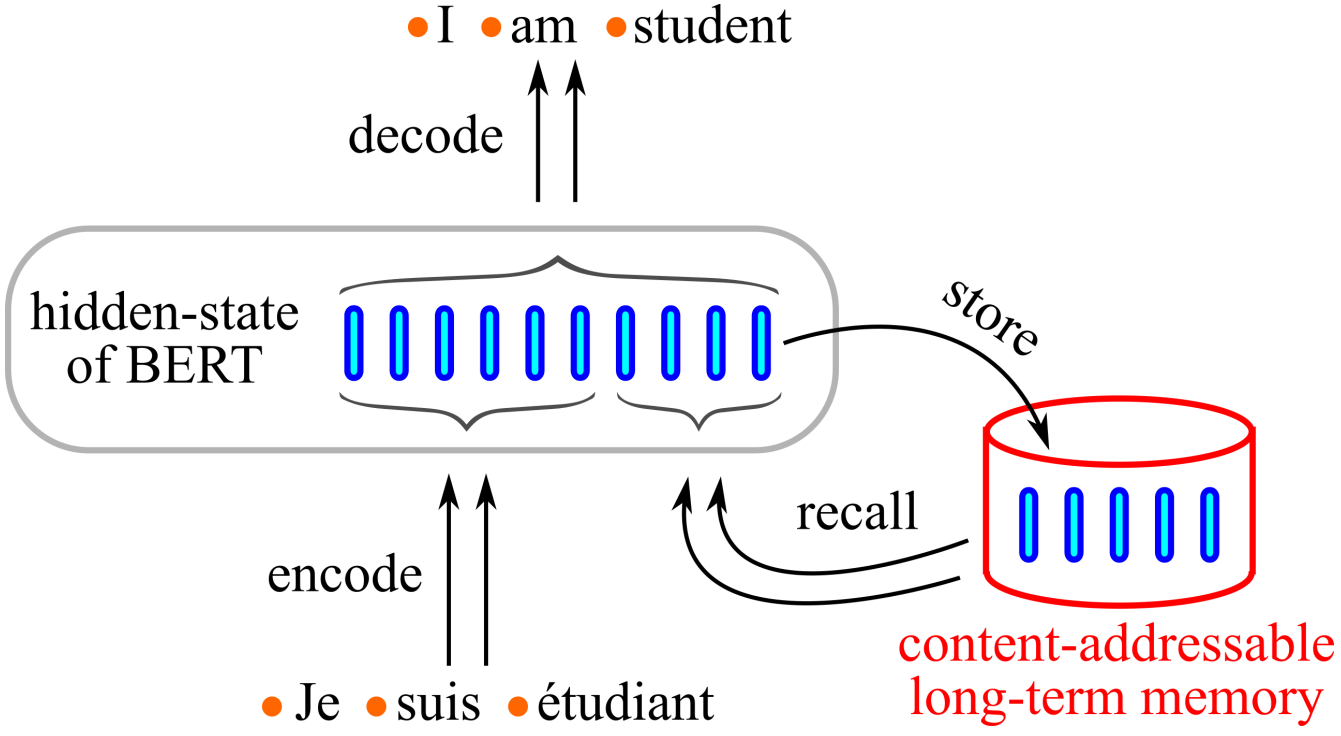
$$\boldsymbol{x}_1 \wedge \boldsymbol{x}_2 \wedge \boldsymbol{x}_3 \vdash \boldsymbol{y}_1 \quad , \quad \boldsymbol{x}_1 \wedge \boldsymbol{x}_4 \wedge \boldsymbol{x}_5 \vdash \boldsymbol{y}_2 \tag{15}$$

BERT 为什么成功？

- 6 层的 BERT 只有 $512 \times 6 = 3072$ 个 head，甚至 $8\times$ multi-head 也只是 24576 个，如果每个 head 对应一条 逻辑 formula，这是很少的数目，为什么如此少量的 logic rules 能做到非常成功的效果？
- 我的解释是：BERT 的高层 representation 是一些有 “high-level” 意义的命题，就像在视觉中，高层特征 代表一些复杂的物体
- The embedding of high-level propositions in vector space may be “semantically dense”, meaning that slight changes in the vector position may convey many different meanings
- 由於 logic rules 是以 6 层的 hierarchy 组织而成，这些 rules 具有 “deep learning” 的特性

Content-addressable long-term memory

- Content-addressable memory 的想法来自 Alex Graves *et al* 的 Neural Turing Machine [2014]
- 以前 BERT 的隐状态 没有逻辑结构，我们不是很清楚它的内容是什么；逻辑化之后，BERT 内部的命题可以储存在 长期记忆 中：



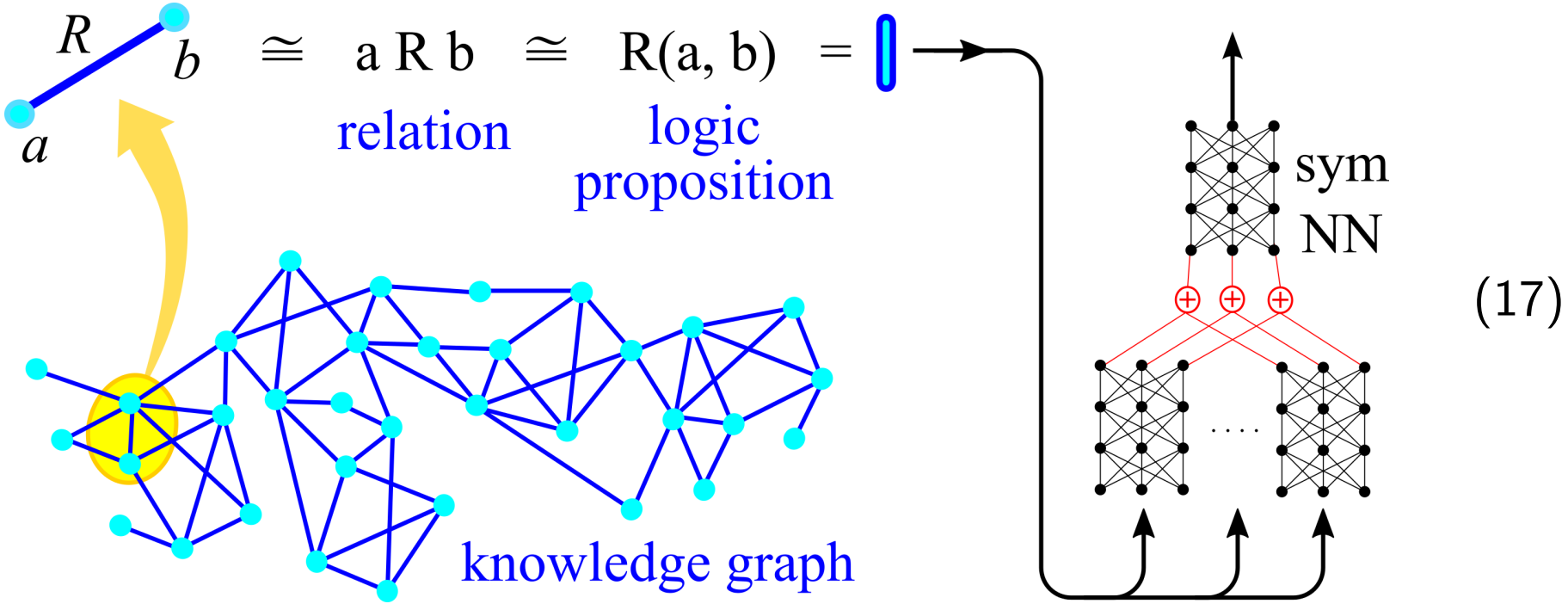
(16)

- 命名：Knowledge-Enhanced Reasoning with Memorized Items
- 这种系统 已非常接近 strong AI，而这是有赖 逻辑化 才能做到的

- 例如：「太阳是热的」、「水向下流」是经常正确的命题
- 但这些知识很多是 implicitly 存在於 rules (matrix weights) 之中
- 也有些知识是 explicit 的，例如：「猫是哺乳类动物」、「吸烟可以致癌」
- 逻辑化理论提供一种 诠释 logic rules (weights) 的方法
- 也可以将 weights 存进 content-addressable memory

知识图谱 (knowledge graphs)

- 知识图谱 不能直接输入神经网络，它必需分拆成很多 edges，每个 edge 是一个 关系，也是一个 逻辑命题 ；也可以说 “graphs are isomorphic to logic”



- 而这些 edges 似乎必需用 symmetric NN 处理，因为它们是 permutation invariant
- 逻辑化 建立了 知识图谱 和 BERT 之间的一道桥梁

对 逻辑主义 的质疑

- 很多人怀疑：人脑真的用 逻辑 思考吗？
- 其实我们每句表达的 语言，都是逻辑形式的 (logical form)
- 直觉认为，人脑 构造一些 models，再从 model 中「读出」一些结论
- 例如给定一个描述：「已婚妇人出轨，用刀刺死丈夫」

(18)

- 那么 妻子穿著什么衣服？衣服什么颜色？这些都是 臆想 出来的细节，是不正确的
- 这个 model 可以有哪些细节？答案是：任何细节都不可以有，除非是 逻辑上蕴含的，或被 逻辑约束
- Model 本身可以是一些 抽象的逻辑命题 构成的，这也合理；反而，一个有很多感官细节的 model 并不合理
- 其实人脑可能 比我们想像中 更接近逻辑

References

欢迎提问和讨论 ☺️

[1] Alex Graves, Greg Wayne, and Ivo Danihelka. "Neural Turing Machines". In: *CoRR* abs/1410.5401 (2014). arXiv: 1410.5401. URL: http://arxiv.org/abs/1410.5401.

[2] Qi et al. "Pointnet: Deep Learning on Point Sets for 3D Classification and Segmentation". In: *CVPR* (2017). https://arxiv.org/abs/1612.00593.

[3] Zaheer et al. "Deep sets". In: *Advances in Neural Information Processing Systems* 30 (2017), pp. 3391–3401.